The International Journal of Advanced Manufacturing Technology
https://doi.org/10.1007/s00170-019-03621-5

ORIGINAL ARTICLE

Modeling the Tennessee Eastman chemical process reactor using bio-inspired feedforward neural network (BI-FF-NN)

Alaa Sheta[1] · **Malik Braik**[2] · **Heba Al-Hiary**[3]

Received: 8 September 2018 / Accepted: 19 March 2019

Abstract
This study explores the application of bio-inspired algorithms (BIAs) in training artificial neural networks (ANNs) in the area of the nonlinear chemical process modeling. Motivated by the increasing complexity and operational efficiency of chemical processes, the need for schemes that can improve model identification of highly nonlinear systems is demanded. As a case study, the Tennessee Eastman (TE) chemical reactor problem is considered. The key issue is to find a particular architecture for an ANN that best model the TE chemical process. We propose the use of BIAs of bat algorithm (BA), firefly algorithm (FA), and artificial bee colony (ABC) algorithm as mechanisms to automatically update the synaptic weights of ANN. These algorithms were conducted to increase the ability of ANN to adapt the dynamic aspect of the TE process and prevent trapping into a local optimum. The proposed modeling framework was devised with extensive experiments and statistical analysis to illustrate the usability and suitability of the entire modeling and identification procedures. The results were explored with a discussion using the mean square error (MSE) and the variance account for (VAF) criteria to assess the degree of identification performance of the TE case study model. The reliability of the presented models was compared with previously developed models for the same subsystems of the TE reactor using competitive, intelligent approaches. The comparative results show that the proposed approach provides superior modeling performance and outperforms its competitors.

Keywords Tennessee Eastman process · Neural networks · Bat algorithm · Firefly algorithm · Artificial bee colony algorithm

1 Introduction

Process industry has ever played an important role in the global economy. Further, data-based models and applications in plant-wide operations have caught considerable interest in academia and industry. With the development of chemical process plants, they have become more complex and onerous to control. In this context, the increase of global competition in controlling and monitoring industrial processes has led to featuring new modeling schemes for design and analysis of non-linear industrial processes [31]. The goal is to meet the higher demands of process safety, product quality, product efficiency, consistency, and even the productiveness of all running equipment. Industrial modeling techniques have been growing in acceptance and have further received a significant development in both the academic community and practical sites.

Recently, a considerable effort has been devoted to the evolution of control and monitoring models for linear and nonlinear systems with various kinds of assumptions and identification scenarios [6, 11]. At present, the fruitful achievements in process modeling and identification have become more integrated into process control systems and have begun to play a remarkable role for industrial and chemical plants [33].

✉ Alaa Sheta
alaa.sheta@tamucc.edu

Malik Braik
mbraik@bau.edu.jo

Heba Al-Hiary
hhiary@bau.edu.jo

[1] Department of Computing Sciences, Texas A& M University-Corpus Christi, Corpus Christi, TX, USA

[2] Department of Computer Science, Al-Balqa Applied University, Salt, Jordan

[3] Department of Computer Information System, Al-Balqa Applied University, Salt, Jordan

Published online: 04 April 2019

A broad range of applications in scientific and engineering fields relies on the derivation of mathematical models for monitoring and controlling purposes of real industrial processes that pose a challenge to constructing such mathematical models [30]. Applications such as model optimization and process control require well-aimed simulation models to assist improving production quality, increasing profits, and reducing computational burden.

Automation of industrial chemical processes demands the development of several mathematical models to increase the possibility for the simulation and quantification of complicated industrial operations. This requirement poses a major request to develop high-quality modeling methods for nonlinear industrial processes to ensure both the required productivity and service quality. Creating a mathematical model for the process is important to establish a relationship between driven input and output variables. There are two popular approaches to dealing with such modeling issues.

1.1 Conventional empirical modeling methods

Conventional empirical modeling methods rely on the construction of an automated physical model based on fundamental physical data recorded in experimental or industrial process characteristics. The physical modeling process is one of the most attractive and challenging issues in control theory and its applications. Some factors make traditional modeling a tedious and time-consuming task which may lead results that do not bode well, particularly if there is a lack of precise, formal knowledge about the system or even if it has a high degree of uncertainty. These non-model-based approaches to the representation of models are fragile and might fail with modest changes in system linearity or if the system suffers from unstable or noisy data, making it difficult to be derived from the analysis of the first principles. Also, these empirical models may generate inaccurate results for many cases that do not represent the data well, and sometimes the developed model is not understandable [10, 20].

1.2 Model-based methods

Model-based methods are about deriving a mathematical relationship model between the observable and the actual data of a system. The goal to minimize the difference between the target data and the original data output. In doing so, a set of process variables is used as inputs for the model [5, 29]. Thus, there is a challenge in how to create a model for a real complex process with sufficient appropriateness that allows the model to identify the data measurements efficiently. Extracting a suitable model structure for a chemical industrial process is necessary for model-based identification experiments.

Optimal design in the industry often relates to multiple design objectives under complex and highly nonlinear constraints. Different objectives often conflict each other, and sometimes, truly optimal solutions do not exist, sometimes, some compromises and approximations are needed [12]. In addition to this complexity, a design problem is subject to several design constraints, limited by design standards, material properties, and even the optimal utilization of available resources and costs [12].

Despite being a fairly old process, the Tennessee Eastman (TE) chemical reactor [9] process remains an important tool and a well suited throughout all disciplines of system theory. The process involves comparative studies or verification of algorithms such as system identification, evaluation of control systems and monitoring processes. However, modeling and identification methods of the TE process have posed a challenge to the control community because it is nonlinear and has unstable conditions.

This paper presents a model-based method to model the dynamics of a real plant-wide chemical process reactor using bio-inspired algorithms (BIAs) [19] in training artificial neural networks (ANNs). ANN learning approach focuses on the automatic evolution of the synaptic weights of the ANN. The BIAs has a Swarm intelligence (SI) nature. SI algorithms work based on the collective behavior of a population of cooperating agents that target one mission to optimize a problem solution [32]. The population of agency can show a form of self-organization behavior that can find a pathway to the best solution of a problem among huge possibilities of solutions. SI-based algorithms have become very popular due to their ability to solve a wide range of optimization problems in computer science and engineering [7, 21, 22].

The swarm intelligence (SI) optimization algorithms are selected for this learning process include bat algorithm (BA), firefly algorithm (FA), and artificial bee colony (ABC) algorithm. The aim involves getting a particular structure of an ANN with an optimal value of the weights. Consequently, the resulting structure can be used to create a mathematical identification model using the input and the output dataset series for the industrial case study problem. It is anticipated that this will make it possible to compare alternative models of the case study as a second objective. A third objective is to assess the developed models using a set of performance measures.

The remainder of this paper is structured as follows. Section 2 briefly describes the challenges concerned with modeling the TE process. Section 3 briefly describes the TE simulation process, case study. Section 4 introduces a description of the ANN and the challenges with traditional ANN training methods, thoroughly illustrating the selection of the model structure. Section 5 is devoted to present the proposed bio inspired algorithms that are used to

train the ANN to build the structure of the models whereas Section 6 describes the proposed criteria we use for evaluating the proposed approaches. Section 7 describes the experimental training and testing datasets and also highlights the preliminary pre-steps of model development. The simulation results together with the performance analysis of the obtained model results are shown in Section 8. Additionally, Section 8 addresses comparative studies with concluding comments and outlooks for further work in Section 9.

2 TE process challenges

Development of reliable and accurate models to simulate the outputs of complex industrial processes is necessary to generate a good approximation of the output variables of industrial processes. TE process has been broadly applied by process control and monitoring community as a data source to compare various approaches and assess the efficacy of process monitoring techniques and process control researches [6, 8, 14, 18, 37]. However, modeling and controlling methods of TE process have encountered a challenge because of the many uncensored effects on the TE process. The variations in the parameters in this process are a secondary challenge. Ricker and Lee [24, 25] derived a physical model of the TE and used this model to develop a nonlinear model predictive control algorithm.

There are many existing methodologies of commonly used modeling and identification procedures for the TE plant process. However, it is not practical to review all research statuses with various aspects that have been reported in the literature for monitoring, modeling, and controlling of the TE process. Here, we selected a set of various intelligent and metaheuristics algorithms that were introduced in literature for modeling the subsystems of the TE chemical reactor [2, 3, 11, 28]. The sub-models considered in their proposed approaches include the reactor level, reactor pressure, reactor cooling water temperature, and the reactor temperature. These approaches are related to our work, and it is easy to compare results between our approach and these approaches quantitatively.

Sheta [28] has applied Fuzzy Logic (FL) by Takagi-Sugeno fuzzy models to generate a set of fuzzy rules to model the behavior of the four subsystems of the TE reactor. His proposed models had good approximation capabilities. Al-Hiary and Sheta then presented a reasonable model structure for the TE reactor using Neural Network Auto-Regressive eXternal (NNARX) model [3]. Then, an Adaptive Neuro-Fuzzy Inference System (ANFIS) was formalized to predict the future actions of the output variables derived from the TE system [2]. Recently, Faris and Sheta [11] studied the behavior of the TE reactor by exploring the modeling and identification processes using Genetic Programming (GP) approach. They had constructed a sensible mathematical formulation model that highlighted that GP can capture the important characteristics of the data with relatively simple analytical formulas.

The authors [2, 3, 11, 28] are using the VAF measure to identify the degree of performance of the developed models of the TE reactor. The modeling schemes described above have achieved respectable progress in modeling the subsystems of the TE reactor. Each method has a special ability to approximate the dynamics of the TE subsystems. Each of the reviewed methods performed well for the cooling water temperature and the temperature cases considered, and each fell short of what was required in the reactor level and pressure. Thus, each model will be learned to capture the dynamic aspect of the reactor cooling water and temperature subsystems accurately and has not acted well in identifying the reactor level and pressure subsystems. The above models reported low VAF values and poor assessment results for the reactor level and pressure. Low VAF is considered as a big limitation of the above methods.

These approaches may be particularly sensitive to noise levels, as they did not behave well with the variation and noise that present in the TE dataset. Thus, we hindered the reliability of their developed models. Therefore, it is practical to use a filtration process to reduce the variations in data and detect the faults that affect data quality. Thus, modeling approaches require a pre-processing stage to obtain a richer dataset. Although the researches above have reported excellent results, we believe that there is still room to make further improvement on modeling this process, especially for the reactor level and pressure datasets. In this work, we compare the performance of three optimization methods in training ANNs. Specifically, the optimization techniques: BA, FA, and ABC are evolved to adjust the synaptic weights of ANNs to model the TE chemical process.

3 Tennessee Eastman process chemical description

Tennessee Eastman process is a test setup of a plant-wide open-loop unstable problem referred to as the TE industrial challenge process [25]. The TE process, as presented in [9], is a highly nonlinear chemical process created by an actual plant system, with slight changes made to protect the identity of the reactants and products. This system is a well posed benchmark problem for analysis and control design in the process control community for a hypothetical chemical plant created by the Eastman Chemical Company [9]. Thus, we can provide a realistic simulation of a chemical

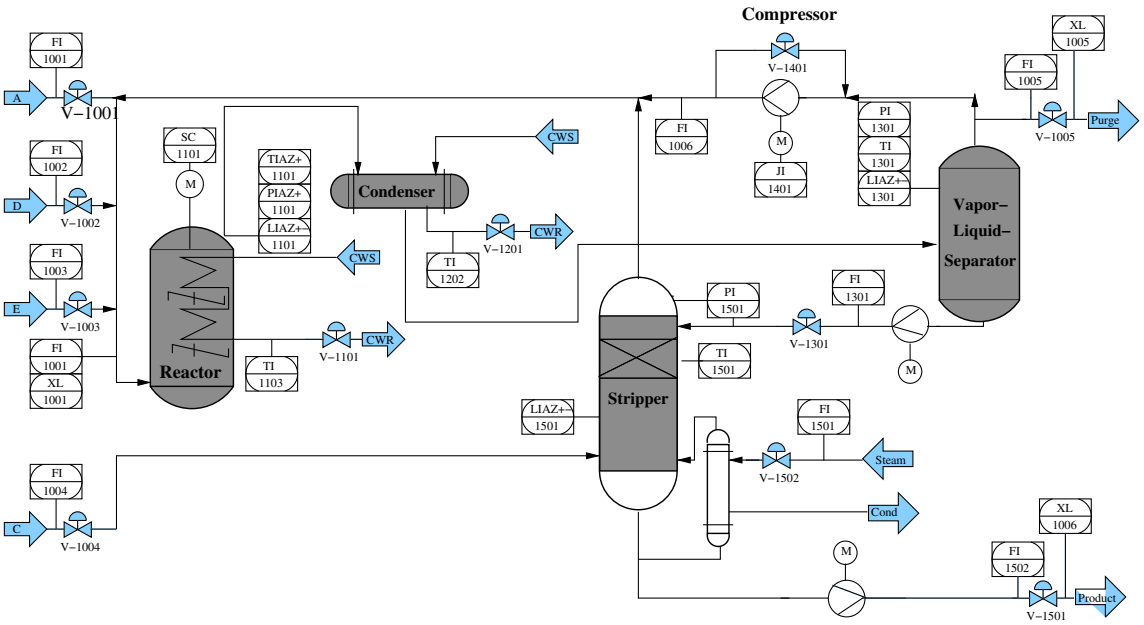

Fig. 1 Schematic flowsheet of the Tennessee Eastman process [4]

industrial process and evaluation of academic research, analysis, control, and monitoring methods. This process consists mainly of five major operating units: a two-phase reactor, a partial product condenser, a recycling compressor, a vapor-liquid separator, and a product stripper.

The reaction process has eight components, including four gaseous reactants (A, C, D, and E), two liquid products (G and H), an inert component (B), and a by-product (F). The reactant components A, C, D, and E along with the component B are fed to the reactor where the liquid products G and H are produced together with a byproduct, F. The two-phase reactor consists of a flash separator, a recycling arrangement, and two additional by-product reactions. Figure 1 represents a flowsheet schematic diagram of the TE process. The nonlinear dynamics of the plant are mainly due to chemical reactions within the reactor.

3.1 Input and output data of the TE system

A large number of interacting process and manipulated variables have been integrated into the TE model, making it a significant plant-wide control problem [15]. According to academic standards, the TE problem is quite large: it provides a total of 41 measurements as available output measured variables, 50 states, and 12 valves available for manipulated variables (input variables). The 41 measurements are a mixture of continuous and sampled measurements. They are used for controlling issues. Each measurement is corrupted by additive noise where the statistical characteristics of the noise are unknown. Of the measured values, 22 values monitored continuously. The other 19 measured values are retrieved from gas chromatographs (three analyzers) that give information only at certain intervals. The analyzers provide information about chemical components in the reactor feed, the purge gas, and the product with sampling delays of 6 min, 6 min, and 15 min, respectively.

The 50 states of the model closely correspond to the actual plant operation which represents the significant system dynamics. The input variables allow the designer 12 degrees of freedom with nine flow valves, two temperature control valves and control of the reactor agitation rate [23]. The 12-manipulated variable, referred to as the agitator speed of the reactor stirrer, is not used in the controller structure and does not change during the workflow of the plant. Therefore, the symptom input vector of the fault diagnostic scheme has 52 components, consisting of 41 measured and 11 manipulated variables.

The simulated process can produce normal operating condition as well as 21 faulty conditions and generate simulated data at a sampling interval of 3 min. For each case, no matter normal or faulty condition, training, and testing datasets were generated by simulating the TE process. The simulation results of the TE process model differ with a selection of the solver used in the simulations.

Figure 2 shows that the choice of solver has a notable effect on the simulation results. The simulation was performed as an open-loop and disruptions of the type "random variation" which affects the reactor cooling water inlet temperature and the formation of the components of stream 4 (see [9]). To maintain the reactor pressure for as

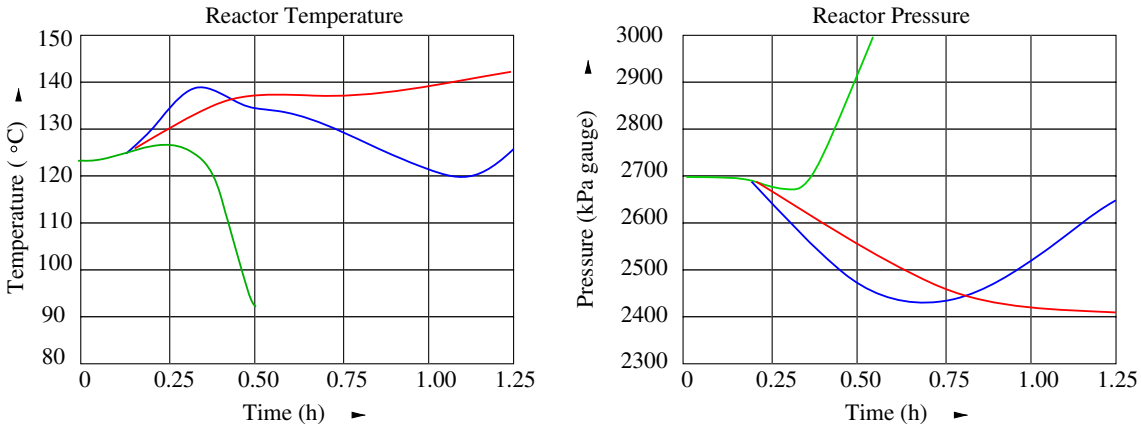

Fig. 2 The simulation results of the reactor temperature and pressure reactor using the main existing model

long as possible under its shut-down constraint of $3000kPa$, the location of the reactor cooling water outlet valve was given a slope of $-\ 8\%/h$. The training sets were used to construct a statistical predictive model and the testing datasets used to estimate the accuracy of the model.

To simplify the demonstration, only the reactor unit shown in Fig. 3 is considered. The reactor consists of 12 variables: feed mole-fractions X_A, X_B, X_C, X_D, X_E, X_F; level L, pressure P, coolant temperature T_c, temperature T, feed flow F and coolant valve position V_c. These variables are presented in Tables 1, 2, and 3. Table 1 summarizes the selected candidate outputs of the TE reactor for stabilization control, Table 2 summed up the manipulated variables of the TE reactor, and Table 3 shows the process measurements of the reactor feed analysis.

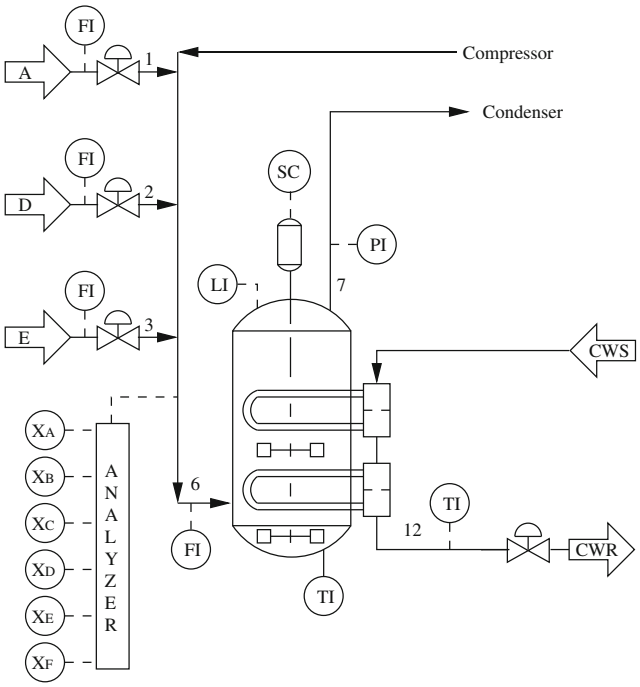

Fig. 3 Tennessee Eastman reactor

The level L, pressure P, coolant temperature T_c, and temperature T are measured outputs. Feed flow F and feed mole-fractions $X_A, \ldots, X_F$ are measured inputs, and the coolant valve position is a manipulated input. At most, five of the six mole-fractions are independent. The entire system is simulated with nine control loops operating with static setpoints. Three variables, temperature, pressure, and level are controlled with their assorted setpoints. The temperature is controlled by manipulating the coolant valve in a cascade loop where the reference for the coolant temperature is varied within the loop. Because of the nine fixed setpoints, only three of the inputs to the reactor are independent, including the coolant valve. Thus, one of the mole fractions and feed flow control the remaining mole fractions. Be means of modifying the three setpoints; the additional control constraints have been eliminated.

4 Artificial neural networks

Researchers from many scientific disciplines have designed ANNs to be a vigorous mathematical tool to solve a diversity of practical problems like pattern recognition, prediction, medical imaging, speech recognition, and control. Among the many NN architectures proposed, feedforward neural networks (FFNNs) have been broadly

Table 1 Candidate outputs for stabilization control of the TE reactor unit

Variable name	Base case value	Unit
Reactor level	75.000	%
Reactor pressure	2705.0	kPa gauge
Reactor temperature	120.40	°C
Reactor cooling water	94.599	°C

Table 2 Manipulated variables in the TE reactor unit problem

Variable name	Base case value	Unit
A feed flow	0.25052	$kscmh$
D feed flow	3664.0	kgh^{-1}
E feed flow	4509.3	kgh^{-1}
Reactor cooling water flow	93.37	m^3h^{-1}
Reactor feed rate	42.339	$kscmh$

used in many domains because of their ability. For example, FF-ANN can (1) provide models for a wide range of natural phenomena that are hard to deal with using classical parametric techniques and (2) can estimate complex nonlinear mappings directly from input samples.

The research interest in ANN relates to its appealing properties it exhibits such as adaptability, learning ability and its capability to generalize. The ANN consists of a set of layers and can also use linear activation functions; the first and last layers are called the input and output layers, respectively and the remaining layers are called hidden layers. There are two basic components in ANN. They are neurons that are interrelated functional elements, and links which have weighting parameters to clasp the interconnections between the neurons. Each neuron receives stimulation from other neurons, processes information, and produces outputs. Neurons are categorized into input, output and hidden neurons. The weighting parameters of the ANN should be initialized before the training process. The training process aims to produce the desired output when applying a set of input to the network; the training process is mainly undertaken using the back propagation (BP)–based learning [26]. During the learning process, the weights must be updated iteratively in a systematic manner. Once the training process completed, the weighting parameters remain fixed throughout the use of the FFNN as a model. In training, the inputs given to the system are fed into ANN, and the output is also fed back to the FFNN which serves as a target pattern. The estimated output is compared with the target value to obtain the error value. This value is then used by the training algorithm to reduce the error by varying the weights at each iteration. Upon completion the learning process (aka the training stage), it is compulsory to evaluate the generalization capabilities of the ANN using samples of the problem, different from those used during the training phase. Finally, the ANN is expected to be able to classify the patterns from a specific problem with acceptable accuracy during the training and testing phases. Conventionally, all the weights and biases of the FFNNs need to be adjusted. Thus, there is a dependency between different layers of weights and biases.

Table 3 Sampled process measurements of the TE reactor unit

Component	Base case value	Unit
A	32.188	mol%
B	8.8933	mol%
C	26.383	mol%
D	6.8820	mol%
E	18.776	mol%
F	1.6567	mol%

4.1 Challenges with traditional ANN training algorithms

Design of an ANN model that is appropriate and effective to model a nonlinear challenge process is a complex task and its performance depends on the model structure, the input and output data driven, the quality of the process model, the model design, the optimal parameter selection method, the selected transfer function and the training algorithm used to adjust the synaptic weights. The training algorithm may be the most important factor among all because it is strongly related to the ability of the proposed training algorithm to identify the model parameters that fit with the prescribed data set. Traditional gradient descent-based learning algorithms, such as BP and it variant Levenberg-Marquardt (LM) method, were used mainly in the training of multilayer FFNNs.

These algorithms were preferred to use by some researchers [3] in training ANN due to their merits such as their efficient implementation, robust in parameters refinement and speedy convergence compared to other methods. These methods assess the error in the network's decision and propagate the error to the weights throughout the network. This can be considered as one of the main obstacles since the search for optimal weights depends strongly on the initial weights, and if they are near local minima, the algorithm may remain trapped in an undesirable solution; that is, they will be far from the optimum or the best solution and be susceptible to converge at local optima. Moreover, most of these algorithms cannot explore multimodal and non-continuous surfaces and they may sometimes resulting in unexpected performance or often resulting in multiple local optima, especially when applied to complex optimization problems. Although reasonable performance has been obtained in certain well-constrained environments for solving a number of real world problems when the networks are trained by BP or LM [2], these gradient-based learning algorithms are relatively slow in learning than required. These learning algorithms may suffer from inappropriate learning steps or may also easily stumble in local minimums. Moreover, the activation functions used in these methods need to be differentiated.

Many authors have handled the problem of learning BP in various ways using a better energy function [27], selecting a dynamic learning rate and adjusting the optimization strategy and/or utilizing adaptation principles [13]. Most of the proposed variations to improve the BP algorithm involve the use of the learning rate, momentum and gain tuning [38] of the activation function to accelerate the network convergence and avoid falling into the local minimum or by replacing BP with more efficient algorithms. Thus, many iterative learning steps are needed by learning algorithms in order to get better learning performance. Unalike these conventional implementations, we turned our attention to BA [36], FA [34], and ABC algorithm [17] as learning algorithms for single-hidden-layer FFNNs, looking for a new inspiration for modeling nonlinear complex problems that can overcome the handicaps of gradient-based techniques and deal well when modeling a complex industrial problem. Theoretically, these algorithms tend to learn much faster with better generalization performance than conventional gradient-based learning algorithms while avoiding many of the difficulties encountered by gradient-based learning methods such as stopping criteria, learning epochs, learning rate and local minimums. Different from traditional learning algorithms, the proposed learning algorithms not only tend to reach the smallest training error but also obtain the smallest norm of weights. However, the proposed network architectures including ANN-BA, ANN-FA, and ANN-ABC may require more hidden neurons than conventional tuning-based algorithms. Since swarm intelligence algorithms are widely used as global searching methods for optimization, a hybrid of SI algorithm and analytical methods must be promising for network training. The proposed BA, FA, and ABC are known for their ability and efficiency to locate global optimum. In the proposed algorithms, BA, FA, and ABC learning algorithms are used to get an optimal set of synaptic weights of ANN and hidden biases, in which an optimal structural model would be obtained.

4.2 Model structure selection

Selecting an optimal structure model for the TE reactor from a set of candidate models is a difficult task, and it is not trivial because the TE reactor is a nonlinear process. The NNARX model structure depicted in Fig. 4 was selected for modeling the TE reactor four sub-problems. We are taking into account that our proposed model has no past input values. The proposed model has three inputs $u_1(k), u_2(k)$, and $u_3(k)$ represents the flow of the reactor, coolant valve position, and the feed mole fraction, respectively, and the past output value $y(k-1)$.

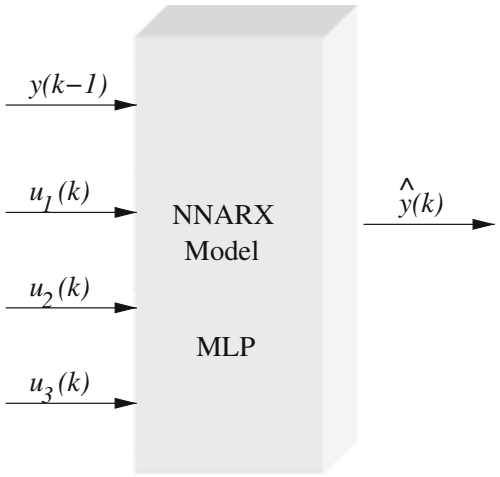

Fig. 4 The proposed NNARX TE reactor model description

The general mathematical class of model vector using output and input datasets for the NNARX model is given in Eq. 1.

$$\phi(t) = [u(t-1), \ldots, u(t-N_A); y(t-1), \ldots, y(t-N_B)]^T \tag{1}$$

where

$$U(t) = [u(t-1), \ldots, u(t-N_A)]$$

$$Y(t) = [y(t-1), \ldots, y(t-N_B)]$$

$$\phi(t) = [U(t), Y(t)]^T$$

where $\phi(t)$ is the function that describes the system model, t is the time instances, N_A is the maximum allowable delay in time for the input u, and N_B is the maximum allowable delay in time for the output y. In this case, the predictor output is given by Eq. 2.

$$\begin{aligned} \hat{y}(t) &= \hat{y}(t|t-1, \theta) \\ &= g(\phi(t), \theta) \end{aligned} \tag{2}$$

where θ represents the model parameters, $\hat{y}(t)$ is the predicted value of the output, and g is the predictor output. The data vector of the proposed TE model can be represented as in Eq. 3.

$$\phi(t) = [y(t-1), u_1(t), u_2(t), u_3(t)]^T \tag{3}$$

where $u_1(t)$, $u_2(t)$, $u_3(t)$ are the three main inputs and $y(t-1)$ is the past output value of the TE model.

5 Proposed bio-inspired optimization algorithms for ANN training

In this section, we describe the three BIAs adopted in this study. Bat [36], firefly [35], and artificial bee colony algorithms [17] were used to train the ANNs to create the models needed to characterize the sub-problems of the TE reactor. These swarm intelligence algorithms, BA, FA, and ABC, are inspired from the collective behavior of social swarms by simulating the so-called SI characteristics of biological agents such as bats, fireflies and bees colony, respectively. Although these swarms consist of relatively unsophisticated individuals, they exhibit coordinated behavior that directs the flocks to their intended targets. These SI-based algorithms have a good acceptance by the artificial intelligence community due to their powerful optimization in solving complex contemporary global problems.

5.1 The bat algorithm

The bat algorithm, nature bio-inspired metaheuristic algorithm, has been developed based on the inspiration of the hunting behavior and echolocation features of micro-bats [36]. Microbats typically use a type of sonar, called echolocation, to detect prey, avoid obstacles, and locate their roosting crevices in the dark. These small bats can emit a very loud sound pulse and listen to the echo that bounces back from the surrounding objects. The pulsations differ in their properties and can be correlated, depending on the species, with their hunting strategies. Some bats use short, frequency-modulated signals to sweep through an octave, while others often use constant frequency signals to locate echoes. The bandwidth signal varies depending on the species and is often increased using more harmonics.

Echo positioning and characteristics of microbats can be summarized as follows: Each virtual bat flies randomly with velocity v_i at position x_i with variable frequency f_i and loudness A_i. When a micro-bat finds its prey, it changes its frequency, loudness, and pulse emission rate. Selection of best solution continues until a certain criterion is met. In this algorithm, a possible solution to the problem is represented by bat positions. The quality of the solution is indicated by the best bat position to its prey. BA uses a frequency tuning technique to increase the diversity of solutions in the population, while using automatic magnification, during the search process, to balance exploration and exploitation processes. This is by simulating the variations in pulse emission rates and loudness of bats. BA has been developed in accordance with three key characteristics of the echolocation process of the microbats.

- All bats use echo positioning to perceive distance, and they also cognize the difference between prey/food and background obstacles.
- The i^{th} bat flies randomly with velocity v_i at position x_i with a frequency uniformly varies from a minimum value, f_{min}, to a maximum value, f_{max}. The i^{th} bat varies its wavelength λ and loudness A_0 to help to search for food. Bats can automatically modify the wavelength of their released pulses and regulate the pulse mission rate $r \in [0, 1]$, relying on the accessibility of their target.
- The loudness of bats ranges from a large positive value A_0 to a minimum fixed value A_{min}.

A number of virtual bats (initial population), P, is generated randomly. Each bat i is associated with a velocity v_i^t and a location x_i^t, at iteration t, in a D-dimensional search space. The three abovementioned rules have been translated into a set of mathematical formulas for frequency f_i, velocity v_i and position x_i for virtual bat i as described respectively below [36]:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{4}$$

$$v_i^t = v_i^{(t-1)} + (x_i^{(t)} - x^*) f_i \tag{5}$$

$$x_i^t = x_i^{(t-1)} + v_i^{(t)} \tag{6}$$

where $x_i^{(t-1)}$ is the current best solution, $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution, and x^* is the current global best locations/solutions, at iteration t, that is located after comparing all solutions among all P bats.

Once a solution is selected among the current best solutions, a new solution is generated locally, at time step t, for each bat using random walk as follows:

$$x_i^t = x_i^{t-1} + \rho A^t \tag{7}$$

where x_i^t is the new bat position at iteration t, x_i^{t-1} is the old bat position at iteration $t-1$, $\rho \in [-1, 1]$ is a random number, and ρA^t is the average loudness of all bats at t time step.

The loudness and pulse emission rates provide a mechanism for automatic control and automatic zoom in the area with promising solutions. The degree of loudness decreases as the bat moves near its prey, while the pulse emission rate is increasing as the bat closes to its prey.

The loudness A_i and the pulse rate r_i for a bat i are updated accordingly at each iteration t. This is shown respectively in Eqs. 8 and 9:

$$A_i^{t+1} = \alpha A_i^t \tag{8}$$

$$r_i^{t+1} = r_i^0 \left[1 - e^{-\gamma t}\right] \tag{9}$$

where α and γ remain constants while running the algorithm. In essence, for $0 < \alpha < 1$ and $\gamma > 0$, the following constraints can be used: $A_i^t \to 0$, $r_i^t \to r_i^0$, as $t \to \infty$

Figure 1 shows the algorithm that summarizes the main steps of the BA used for training the ANN.

5.2 Firefly algorithm

The FA has been developed by Yang [34] as a stochastic, nature-inspired SI algorithm. FA is based on flashing light, attraction characteristics, and social behavior of firefly flocks in the summer sky in tropical and temperate regions. Fireflies search for pray and find mates using a biological scintillation process with varied flashing patterns. There are two fundamental functions of such flashes that are important to attract mating partners and attract potential prey. A firefly tends to be attracted towards other fireflies with high flash intensity. A local attraction, attractiveness, and automatic regrouping are important merits of FA.

The initial positions of fireflies are identified randomly in the search space. In essence, there are two important issues that need to be identified in order to control the flow of optimization of FA properly [35]: the variation of light intensity, I, and the formulation of the attractiveness parameter, β. The light intensity is defined to as the absolute measure of light emitted by the firefly and the attractiveness is a relative measure of light that should be seen in the eyes of the beholders and ruled by other fireflies. The light intensity $I(r)$ varies depending on distance r as shown in Eq. 10:

$$I(r) = I_s/r^2 \tag{10}$$

where I_s is the intensity at the source.

The combined effects of both the inverse square law and light absorption can be approximated using a fixed light absorption coefficient as in the Gaussian form defined in Eq. 11:

$$I(r) = I_0 e^{-\gamma r^2} \tag{11}$$

where I_0 denotes the light intensity of the source, and γ is the light absorption coefficient that can be possessed as constant. The distance between any two fireflies i and j at positions, x_i and x_j, respectively, is expressed as the Euclidean distance, as follows:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{P}(x_{(i,k)} - y_{(j,k)})^2} \tag{12}$$

where $x_{(i,k)}$ and $y_{(j,k)}$ are the k^{th} components of the spatial coordinate x_i and x_j of i^{th} and j^{th} fireflies, respectively, and P denotes the number of fireflies.

Algorithm 1: BA tuning algorithm for the weights of the ANN model.

```
Configure the structure of ANN model
Initialize the positions of all bats, x_i (i = 1, 2, ..., P)
Define pulse frequency f_i at x_i
Initialize the velocity, v_i, pulse rate, r_i, and the loudness, A_i, for all bats.
Evaluate the population x_i using the function (MSE) to find the initial solutions
Find the initial objective values, f_0, for all bats, (i = 1, 2, ..., P)
Find the objective value, f_min, the initial solution, x_i and the initial global solution, x*
while t ≤ number of generations do
    for i ← 1 to P do
        Generate new solutions by adjusting the pulse frequency using Eq. 4, and update velocities using Eq. 5 and locations using Eq. 6
        if rand(0, 1) > r_i then
        Select a solution among the best solutions
        Create a local solution around the best solution that is selected
        Create a new solution by flying randomly
        Evaluate the cost function and select new solutions, f_new
        Update the ANN model
        if f_new < f_0(i) and rand(0, 1) < A_i then
        Update the current best solution x_i
        f_0(i) ← f_new
        if f_new < f_min then
        Update the global best solution x*
        f_min ← f_new
        Increase r_i and reduce A_i using Eqs. 9 and 8, respectively
    Rank the bats and find their current best solution x*
return the optimized ANN model using the global best solution x*
```

It is defined in $2-D$ case as follows:

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{13}$$

The variation of attractiveness, β, of fireflies was handled with distance r as defined in Eq. 14:

$$\beta = \beta_0 e^{-\gamma r^2} \tag{14}$$

where β_0 as a constant represents the attractiveness at $r = 0$.

The movement of firefly i is attracted to a more attractive firefly j at iteration t as specified in Eq. 15:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}(x_j^t - x_i^t) + \alpha_t \epsilon^t \tag{15}$$

where ϵ^t is a vector of random numbers drawn, at time t, from Gaussian distribution in $[0, 1]$ and α_t is a randomization parameter that can be expressed as shown in Eq. 16:

$$\alpha_t = \alpha_0 \delta^t \tag{16}$$

where α_0 is the initial randomness parameter and δ ($0 < \delta < 1$) is essentially a cooling factor.

The basic steps of the FA used for optimizing the ANN weight parameters are shown in Fig. 2.

Algorithm 2: A pseudo code showing a summary of tuning the ANN weights using FA.

Configure the model structure of the Feedforward ANN
Generate an initial population of fireflies, as x_i $(i = 1, 2, \ldots, P)$
Evaluate the initial population using the MSE (cost) function
Evaluate intensity I_i *for firefly* i *at* x_i *using the MSE,* $i = 1, 2, \ldots, P$
for $t \leftarrow 1$ ***to*** *number of generations* **do**
  for $k \leftarrow 1$ ***to*** P **do**
    Evaluate the solutions using the MSE function
    Update the light intensity, I_k
  Rank the fireflies by their light intensity/objectives
  for $m \leftarrow 1$ ***to*** P **do**
    Find the current best solution $fbest$
  Update the light absorption coefficient γ_t *and light intensity* I_t
  Update the current best solution, $fbest$*, and the global best solution, nbest*
  Sort the solutions, $fbest$*, according to the fitness values*
  Update the ANN structure
  for $i \leftarrow 1$ ***to*** P **do**
    for $j \leftarrow 1$ ***to*** P **do**
      Find the distance r according to Eq. 12
      if $(I_i > I_j)$ **then**
      Vary the attractiveness γ_j *according to Eq. 14*
      Move firefly i *to more attractive location at* j
      Evaluate and update the light intensity I_j
      Update the current solution $fbest$
  Rank the fireflies and find their best solution (nbest)
return *the optimized Feedforward ANN-FA structure model using nbest*

5.3 Artificial bee colony algorithm

The ABC algorithm, a population-based swarm intelligence algorithm, is inspired by the intelligent foraging behavior of honeybees [16, 17]. The model of forage selection of honeybees that leads to the emergence of the collective intelligence of bees consists of three basic groups: employed bees, onlooker bees, and scout bees. Each bee represents a position in the search space. Employed bees seek for a particular food source and share information with the recruit onlooker bees about the quality of food sources. Information is the distance and direction from the nest and the profitability of the source. Onlooker bees are responsible for watching the dance area within the hive to select the food source and to obtain information about the quality of food sources from the employed bees, where they choose food sources with better quality. The onlooker bees that choose a food source to explore the neighborhood become employed bees. An employed bee, whose food source is rejected by employed and onlooker bees, will change to scout bees to randomly conduct the environment surrounding the nest to search for new sources of food.

The ABC algorithm generated a randomly initialized population of distributed solutions (i.e., food sources). The food sources are assigned randomly to the employed bees or onlooker bees based on their fitness

First, let the i_{th} food source in the population is represented by $x_i = \{x_{i1}, x_{i2}, \ldots, x_{iD}\}$, so that the employed bees explore the neighborhood of the food sources, x_{ij}. Exploration in the neighborhood is represented as follows:

$$v_{ij} = x_{ij} + r^*(x_{ij} - x_{kj}) \tag{17}$$

where v_{ij} represents the solutions (food sources positions) in the neighbourhood of x_{ij} for the employed bees, x_{ij} and x_{kj} are upper and lower bounds in dimension j, $j \in \{1, 2, \ldots, D\}$, $i \in \{1, 2, \ldots, N\}$, such that $i \neq j$, N is the number of food sources, D is the number of optimization parameters, r^* is a uniformly distributed random number in the range of [0, 1] and indices j and k are randomly chosen indexes.

At the onlooker bee phase, the onlooker bees evaluate the nectar information taken from the employed bees and select the food source, x_i, depending on its probability value, p_i, calculated by:

$$p_i = \frac{f_i}{\sum_{i=1}^{N}(f_i)} \tag{18}$$

where f_i is the nectar amount (fitness value) of the i_{th} food source x_i.

In the scout bees phase, the food source is assumed to be exhausted and abandoned if there is no further improvement through a predetermined number of iterations and the

corresponding employed bees become scout bees. The abandoned food source is replaced by a new food source discovered by the scout.

The main steps of the ABC algorithm in training the ANN are summarized in Fig. 3.

Algorithm 3: The main steps of ABC algorithm in training the ANN.

Configure the structure of the feedforward ANN model
Initialize the bee the solutions (costs) and positions, x_i, $i = 1, 2, \ldots, P$
Define the abandonment counter C_i *for all bees,* $(i = 1, 2, \ldots, P)$
Initialize the current solution, $lbest$*, and the global best solution,* $gbest$
Evaluate the costs and positions using MSE and identify the initial solution, f_0
while $t \leq number\ of\ generations$ **do**
  for $i \leftarrow 1$ ***to*** P **do**
    Produce new solutions (food source positions) using Eq. 17
    Find new bee position and evaluate using the MSE to find new best solution, $lbest$
    if $(lbest \leq f_0(i))$ **then** $f_0 \leftarrow lbest$
    else $C(i) \leftarrow C(i) + 1$
    Update the ANN structure model
  for $i \leftarrow 1$ ***to*** P **do**
    Find the current best fitness (cost) and selection probabilities
  for $m \leftarrow 1$ ***to*** P **do**
    Apply roulette wheel selection process to the fitness values
    Calculate the probabilities P_i *for all solutions using Eq. 18 and produce new solutions*
    Find new bee position and use MSE to find a new cost / solution, f_{new}
    if $(lbest \leq f_0(i))$ **then** $f_0 \leftarrow lbest$
    else $C(i) \leftarrow C(i) + 1$
  for $i \leftarrow 1$ ***to*** P **do**
    if *(there is an abandoned solution for the scout)* **then**
      replace it with a new solution that is evaluated using the cost function
  for $i \leftarrow 1$ ***to*** P **do**
    if $(f_0(i) \leq gbest)$ **then** $gbest \leftarrow f_0(i)$
  Rank the bees and find their best food source solution, $gbest$
return *the optimized ANN-ABC model structure using* $gbest$

6 Evaluation criteria

Model estimation stage describes how the developed model behaves during the identification process. This stage is subject to an MSE measure between the target data and the actual output data. If the identification process converges to an appropriate solution of an optimum MSE value, the process ceases, and the structured model can be used to obtain the expected outputs; otherwise, it will continue iterating until some predefined iterations is completed. This stage is necessary to verify the validity of the model in responding to the actual data, and it is useful to judge the ability of swarm techniques in optimizing the weights of ANN in modeling industrial processes.

The performance of the proposed modeling schemes during training and testing phases is assessed by measuring the estimation accuracy as defined by the difference between the actual values $y(t)$ and the estimated values $\hat{y}(t)$ of the TE output data. These metrics are presented to measure the accuracy of the match between the true output data and the expected output data. If the model fails to perform the specified precision as expected, the modeling process is returned to the start-up phase.

- **Mean of the sum of the Square Error differences (MSE)**

  The first typical metric evaluation measure is MSE between the target output values and the model values, defined as shown in Eq. 19:

$$E = \frac{1}{K}\sum_{t=1}^{K}(y(t) - \hat{y}(t|\theta))^2 \tag{19}$$

  where K is the total number of data values, θ represents the model parameters, and $y(t)$ and $\hat{y}(t)$ are the outputs of the process and the model at index t, respectively.

- **Variance Account For (VAF)**

  The second percentile criterion of fit, VAF measure, as defined in Eq. 20, was also used to test the potentiality of the developed models:

$$VAF = \left[1 - \frac{cov(y - \hat{y})}{cov(y)}\right] \times 100\% \tag{20}$$

  $cov(\cdot)$ is the covariance of the respective data vector.

- **Correlation**

  The correlation measure, $\text{corr}(y, \hat{y})$, between the actual output y and the estimated output $\hat{y}$ with mean values of μ_y and $\mu_{\hat{y}}$ and standard deviations of σ_y and $\sigma_{\hat{y}}$, respectively, as defined in Eq. 21, was considered as a statistical measure to characterize the manner of

reacting the developed models to the behavior of the TE subsystems:

$$\text{corr}(y, \hat{y}) = \frac{\text{cov}(y, \hat{y})}{\sigma_y \sigma_{\hat{y}}} = \frac{E[(y - \mu_y)(\hat{y} - \mu_{\hat{y}})]}{\sigma_y \sigma_{\hat{y}}} \quad (21)$$

where E represents the expected value operator and $cov(y, \hat{y})$ is the covariance between y and $\hat{y}$.

In addition, a statistical analysis was performed using the standard error of the mean to quantify the accuracy of each presented scheme. To calculate the standard error of the mean for a set of data measurements:

1. Calculate the mean ($\bar{x}$) of the data measurements, as described by Eq. 22:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \quad (22)$$

where n represents the total number of data measurements, and x_i represents the i^{th} value of the data.

2. Calculate the standard deviation (σ) of the data measurements, as defined by Eq. 23:

$$\bar{x} = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n - 1}} \quad (23)$$

3. Calculate the standard error of the mean (SE) as:

$$SE = \frac{\sigma}{\sqrt{n}} \quad (24)$$

The error bars of SE can be used to describe the uncertainty in data measurements. SE is a statistical measure used to determine whether the variability between the means of two sets of measurements are statistically significant. When the standard error bars between the possible bands of two means do not overlap, then it can be concluded that the difference is statistically significant, while the difference may not be statistically significant when the standard error bars overlap.

7 Experimental datasets

The model development process requires some necessary preparation steps that must be completed as a pre-step before building the model. The process involves data collection, preparation, and pre-processing. The datasets of the TE two-phase reactor were gathered from a variety of operating conditions. These datasets contained both process input variables and manipulated or controlled output variables. The output parameters for this case study, as described in [9], describe the behavior of the process and show how it responds to various input sets. At any instant, the estimation of the future response of this process depends on both the current input and the previous state of the output variables. The TE reactor considered in this study composes of four subsystems denoted as R_L, R_P, R_C, and R_T. The sub-model R_L corresponds to the reactor level, R_P to the reactor pressure, R_C to the reactor cooling water temperature, and R_T to the reactor temperature.

Typically, the success of modeling processes relies on the amount of data that should be sufficiently large to study the behavior of data well. The presupposes a considerable computational time during training. Thus, it is beneficial to carefully choose a sufficient number of data measurements to training the proposed model so that a qualified model is formed. The data measurements for each subsystem of the TE reactor consist of 300 simulated data samples, available at [1]. The dataset for each subsystem was split evenly into two parts: a training dataset used to train the model and a test dataset used to evaluate the performance of the trained model. Thus, the number of data samples for each input variable and output variable in the training and testing processes consists of 150 samples.

Each of the four subsystems of the TE reactor has three main input variables defined as, the feed flow to the reactor, the coolant valve position, the feed mole-fractions, and the past output values. The input variables for each subsystem were indicated in the scope of the modeling problem as $u_1(k)$, $u_2(k)$, $u_3(k)$ and $y(k-1)$. The output of each subsystem is represented by $y(k)$. Figures 5, 6, 7, and 8 show the three main inputs $u_1(k)$, $u_2(k)$, and $u_3(k)$, and the previous output $y(k-1)$ for each of the four sub-problems. The measurements at past sampling time, $k-1$, were identified as the fourth input variable of the model since the model was constructed using time series data. The reason is, the chemical process variables are always auto-correlated. The inputs of the past output values are likely to have a positive impact in establishing a useful identification system and are likely to improve the performance of the developed TE models.

7.1 Data pre-processing

The data used to construct a model-based for a nonlinear real data must be carefully chosen to ensure that it is rich enough and to avoid falling into an over fitting problem. Peak shaving and smoothing of intensive changes in real data are very important in pre-processing data. Data pre-processing methods should be addressed before the learning process to improve the quality and quantity of raw training patterns to support the acquisition of a high-performance modeling method. These preliminary steps are to be applied to the raw experimental data to extract valid input and output

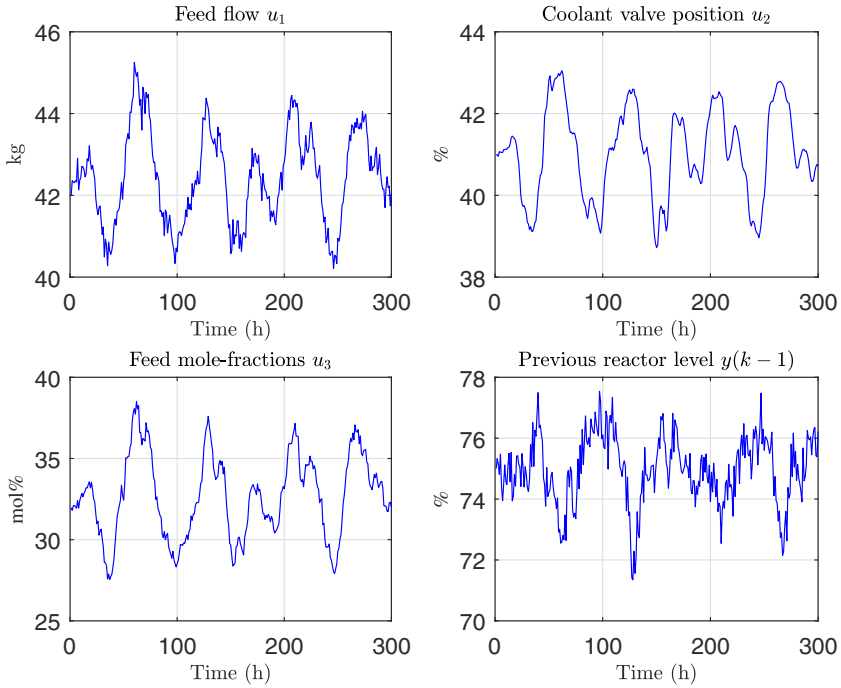

Fig. 5 Input data for the reactor level model

data. The simplest preliminary methods may take the form of data filtering and scale.

7.1.1 Data filtration

In this work, the raw data of the four subsystems of the TE system are smoothed and passed through a first-order digital Butterworth low-pass filter with a bandwidth (cutoff rate) of 0.8 Hertz (Hz) and a sampling rate of 3 Hz. From a signal processing standpoint, a suitable filter should not change or affect the original signal shape while eliminating noise and disturbing signals. This process was implemented beforehand to eliminate the noise posed in the TE data problem to obtain a richer dataset.

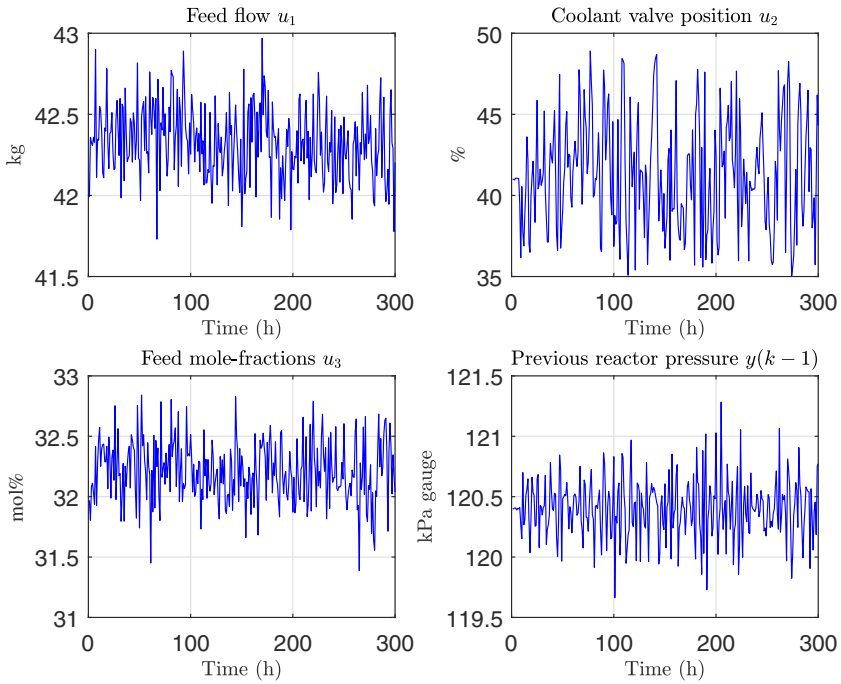

Fig. 6 Input data for the reactor pressure model

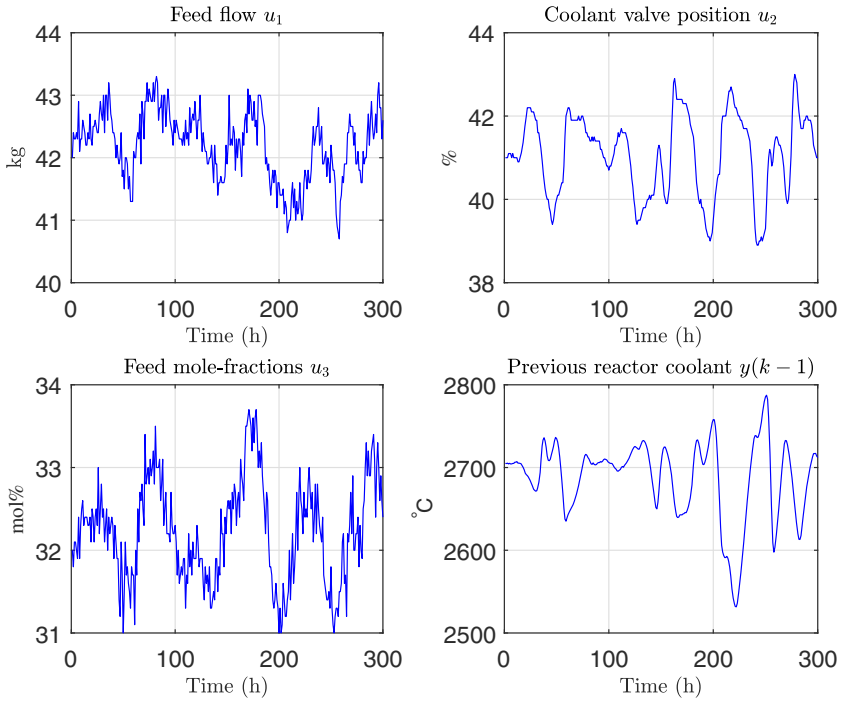

Fig. 7 Input data for the reactor cooling water temperature model

7.1.2 Data normalization

The input and output data of the TE reactor have different ranges, which may result in poor modeling process along with relatively high errors in the identification process. Moreover, the dataset is not normally used directly in the construction of industrial process models because there is a variation in the magnitude of the process variables. Therefore, data normalization should be undertaken as a fundamental step before creating the model. Data scaling should be performed in a fixed range to prevent unnecessary domination of certain variables, banning data

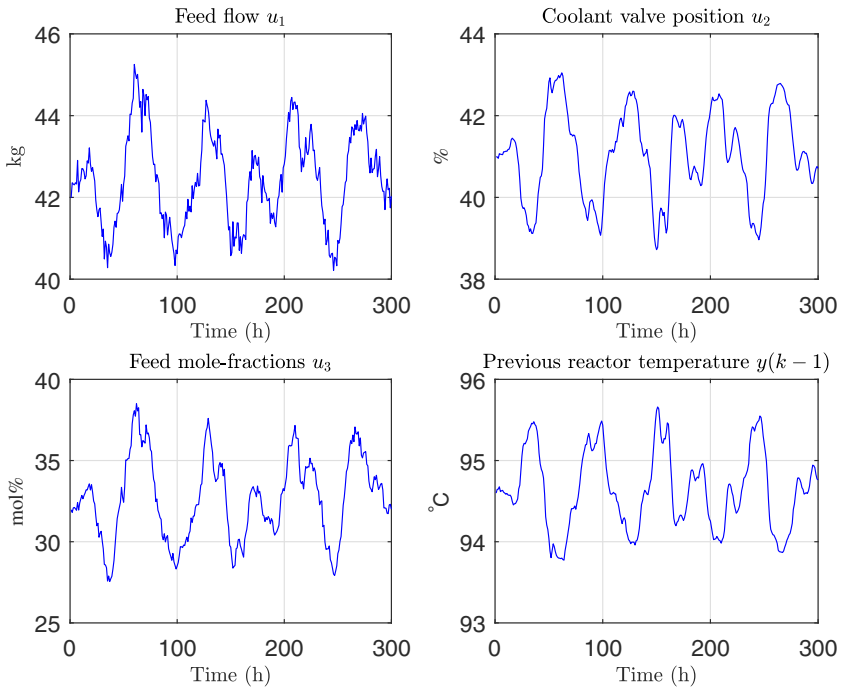

Fig. 8 Input data for the reactor temperature model

of greater magnitude from overriding smaller magnitudes and obstructing the premature learning process. In this study, the original data D was mapped to the normalized data D' as follows:

$$D' = \frac{D - D_{min}}{D_{max} - D_{min}}, \tag{25}$$

where D_{max} represents the maximum values of the original data and D. D_{min} represents the minimum values of the original data, D.

8 Modeling the TE process using BI-FF-NN

The proposed modeling schemes are expected to generate models for the TE subsystems with interpretable structures that relate input and output variables with key parameters selection. The goal of the TE models is to produce an output, $\hat{y}(t)$, that is almost similar to the real output $y(t)$.

8.1 Experimental setup

The Feedforward neural network model was configured with one hidden layer and 15 neurons in the hidden layer, which were accompanied by input and output datasets of the TE reactor. Then, the weights of the ANN were controlled through each of the optimization algorithms of BA, FA, and ABC. The MSE function was used as a fitness function for each proposed training algorithm to investigate the learning characteristics of the model. The control parameters of BA, FA and ABC algorithms and their values are tabulated in Table 4.

The evaluation process stops when the maximum number of evaluations is met. Each model has been experimented several times to ensure that the modeling error converges to a minimum value so that an optimal solution can be found. The population size for each tuned algorithm was set to 40 to assess the performance of the approaches without much variations uniformly.

Table 4 Values of the controlling parameters of the tuned swarm intelligence algorithms

Algorithm	Parameter	Value
	Number of virtual bats	40
	Generations	500
BA	A_0	0.5
	f_{min}	0.0
	f_{max}	2.0
	r^0	0.5
	Number of fireflies	40
	Generations	500
FA	β_0	0.2
	α_0	0.25
	γ	1.0
	Colony size	40
ABC	Generations	500

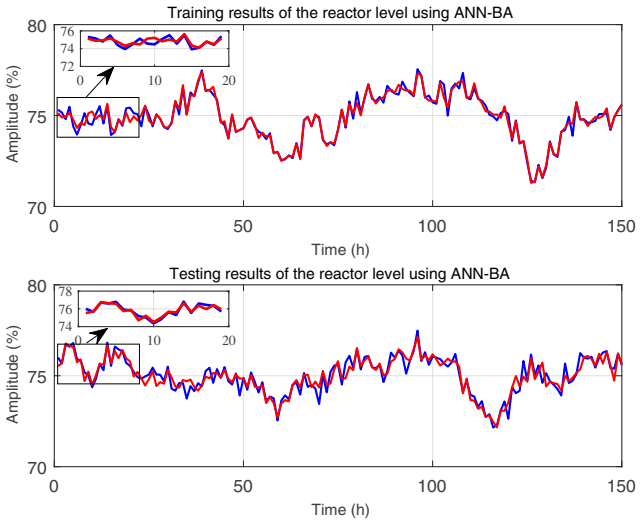

Fig. 9 Actual and estimated response of the reactor level model using ANN-BA

8.2 Simulation results

In the training phase, the conduction weights for all interconnections between neurons are updated based on the proposed optimization algorithm to reach a predefined iteration number or to meet an objective criterion. Through these activities, the structured model learns the behavior of the output response, where each independent model of each sub-process of the TE is learnt with a compact set of

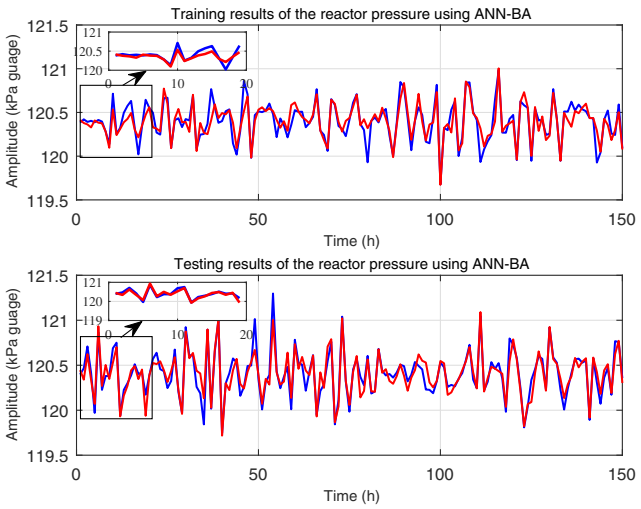

Fig. 10 Actual and estimated response of the reactor pressure model using ANN-BA

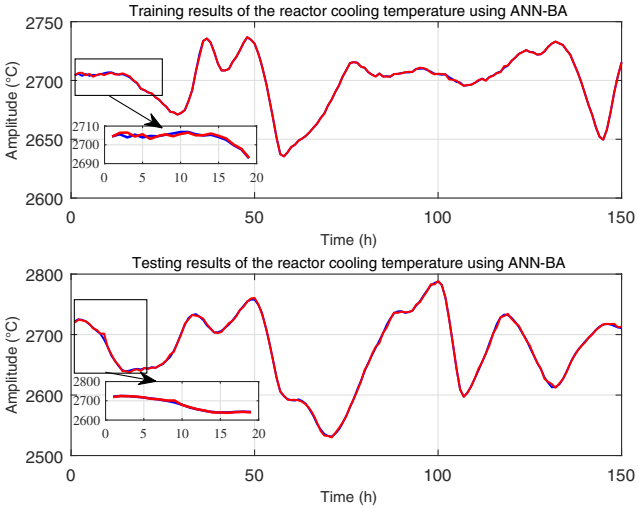

Fig. 11 Actual and estimated response of the reactor cooling water temperature model using ANN-BA

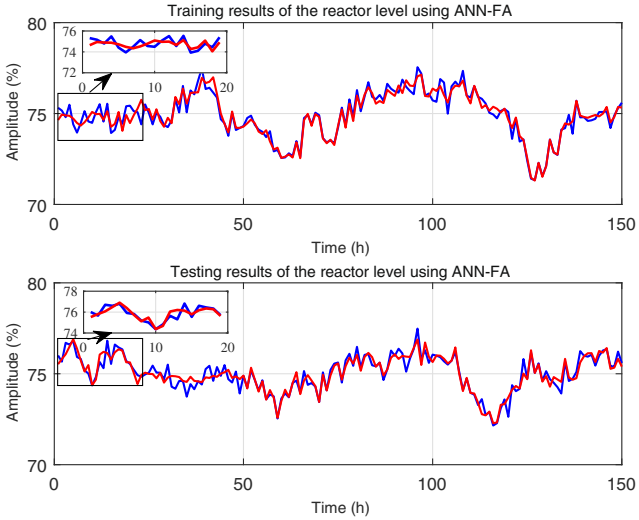

Fig. 13 Actual and estimated response of the reactor level model using an ANN-FA

parameters. The objective function or the mean square error (MSE) was optimized between the network output and the target of each optimization algorithm, during the training process, to measure the closeness between the expected output and the target output. The proposed SI algorithms are expected to avoid local minimum by exploring a large area of the search domain. Each with exploratory search features, making it suitable to optimize the weights of ANNs and provide better scope to get an optimum solution.

The developed models were used to identify the expected output data of an unfamiliar input data that simulates the actual output data. The training and testing datasets each contain input and output data for a subsystem of the TE reactor.The training dataset consisted of 150 data samples of each input variable of each subsystem of the TE process. The dataset used in the evaluation process consisted of 150 unseen samples of each input variable of each subsystem. These testing datasets were distinct from the training datasets.

The simulation results are shown during training and verification stages to illustrate the capability of the modeling scheme. Figures 9, 10, 11, and 12 show the characteristics of the estimated outputs of the reactor level, pressure, cooling water temperature, and temperature compared to the true outputs of each corresponding subsystem of the TE reactor using the evaluated ANN-BA scheme. As a preliminary result, it proves to be effective with a typical quick start.

The performance of the proposed ANN-FA scheme in tracking the real output data of the reactor level,

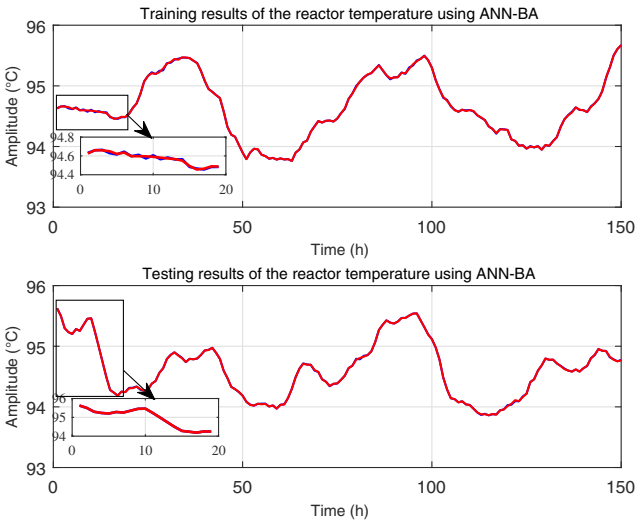

Fig. 12 Actual and estimated response of the reactor temperature model using ANN-BA

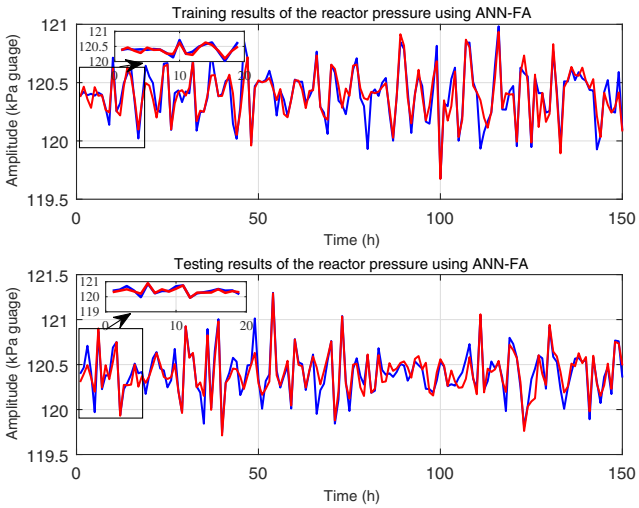

Fig. 14 Actual and estimated response of the reactor pressure model using ANN-FA

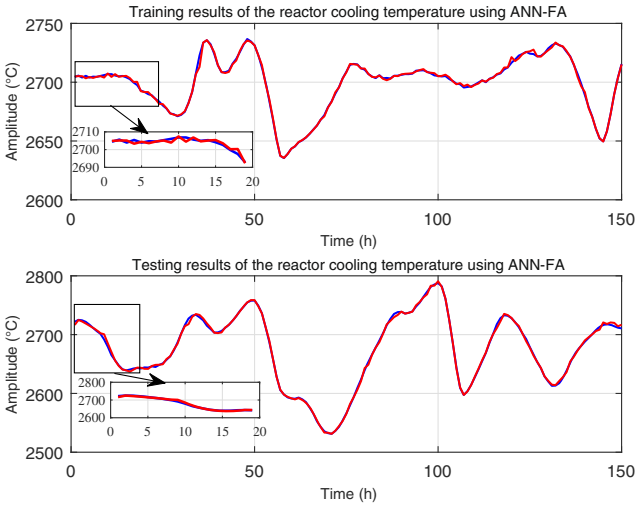

Fig. 15 Actual and estimated response of the reactor cooling water temperature model using ANN-FA

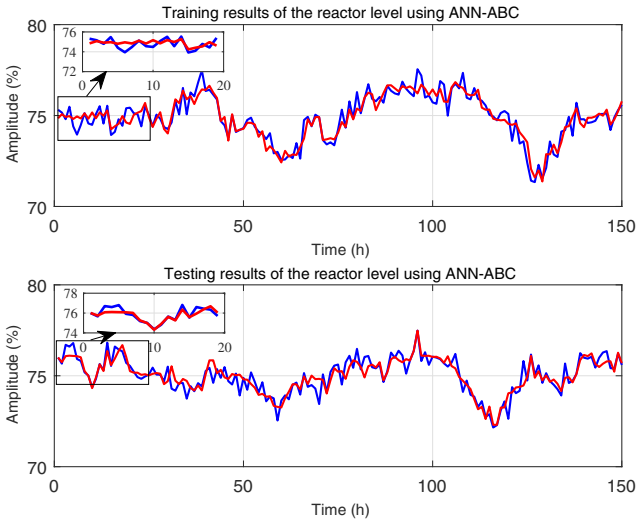

Fig. 17 Actual and estimated response of the reactor level model using ANN-ABC

pressure, cooling water temperature, and temperature of the TE reactor at both training and validation stages, after 500 iterations, are shown in Figs. 13, 14, 15, and 16, respectively.

The data of plots in Figs. 17, 18, 19, and 20 represents the predicted data compared to the actual data for the reactor level, the pressure, the cooling water temperature and the temperature of the TE reactor in both training and verification stages, respectively, where the models are created using ANN-ABC algorithm.

It is observed from the modeling and identification results shown in Figs. 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20 that the predicted data output identified by red plots fit competently to the actual data output as identified by blue plots for each subsystem of the TE reactor, as judged visually. These results appear particularly prominent given a model with a relatively small training dataset. This, in turn, means that refinement and data normalization procedures are valuable to retain the significant aspects of the datasets.

It is observed that the behavior of ANN-BA, ANN-FA, and ANN-ABC is almost similar, although there is a small amount of distinction between the plots which indicates that the difference is non significant.

There is a slight difference between the data plots, and it indicates that the models created using ANN-BA achieved better training than the models created using ANN-FA and ANN-ABC. Hence, the identification results show that the BA is the best learning algorithm among the other evaluated SI algorithms as it yielded an outstanding representation of the future actions of the TE reactor.

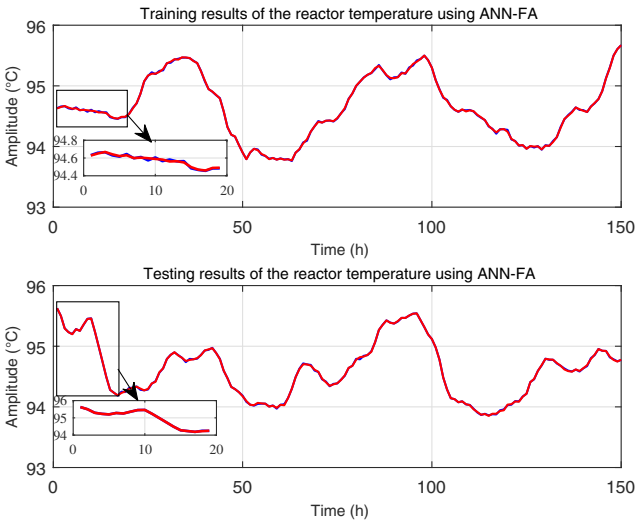

Fig. 16 Actual and estimated response of the reactor temperature model using ANN-FA

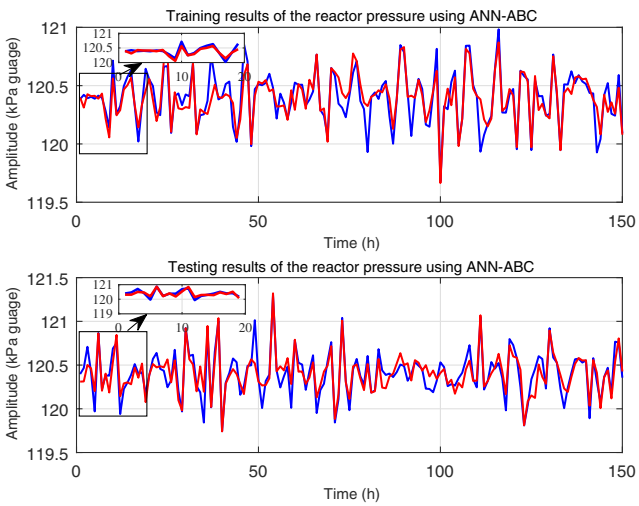

Fig. 18 Actual and estimated response of the reactor pressure model using ANN-ABC

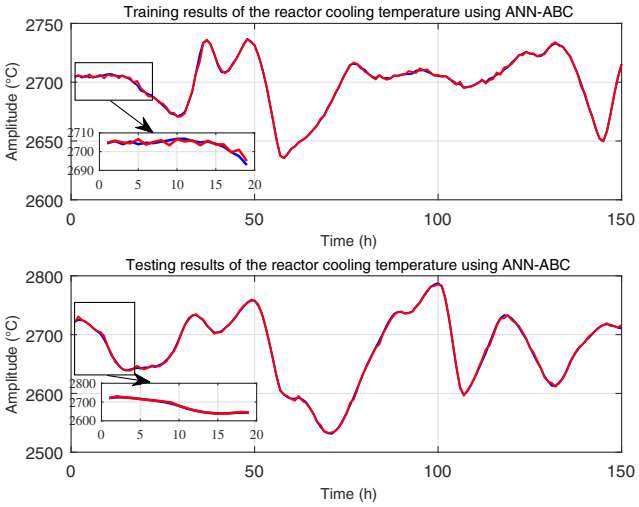

Fig. 19 Actual and estimated response of the reactor cooling water temperature model using ANN-ABC

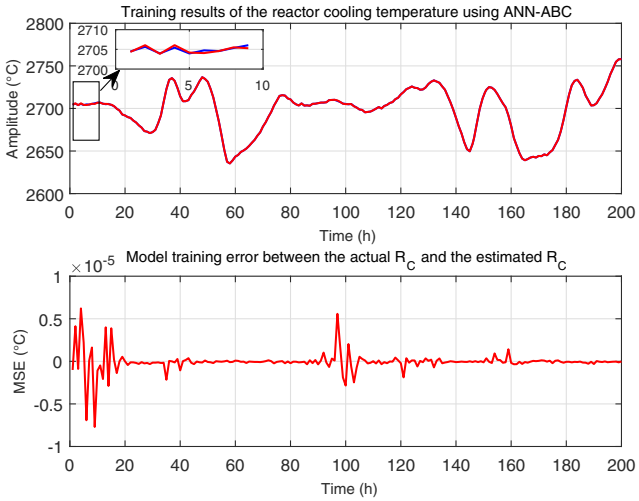

Fig. 21 Training results along with the prediction error between the actual R_C and the estimated R_C using ANN-ABC

To conduct a more comprehensive assessment of the simulation results, a set of experiments was designed to create models for the TE reactor for some illustrative cases for a training set of 200 samples and an independent test set of 100 samples. Figures 21 and 22 show the simulation results in the follow-up of the actual data of the cooling water temperature and pressure subsystems along with the plots of the prediction errors during training and identification phases, respectively.

In Figs. 21 and 22, we show the difference between the actual output data and the estimated output data is not significant. The very small errors perceived in the error prediction plots clarify that the modeling schemes have produced models with a high degree of confidence and predictability.

Figure 22 also shows that the model created with 200 samples is a slightly better representation than the one created with 150 samples as shown in Fig. 10 and suggests that a large training set is used basically to improve the identification process significantly. The error commensurate with the training data set size, it reduces with increasing the size of the training data set and at this stage, it has become very small. Any further improvement in error measuring is likely to require a very large increase in training data set size. It is possible that a large increase in the size of training data set will greatly further minimize the error measure. The error measure may reduce further with more than 500 iterations with a data set of 150 samples.

The convergence curves, of the performance of ANN-BA, ANN-FA, and ANN-ABC, for the identification of

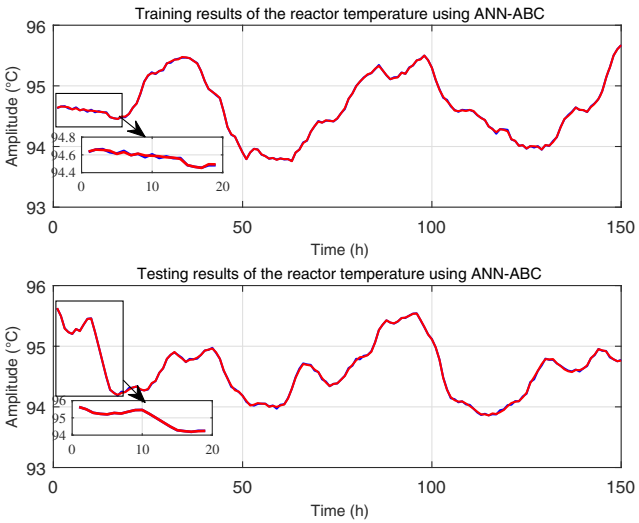

Fig. 20 Actual and estimated response of the reactor temperature model using ANN-ABC

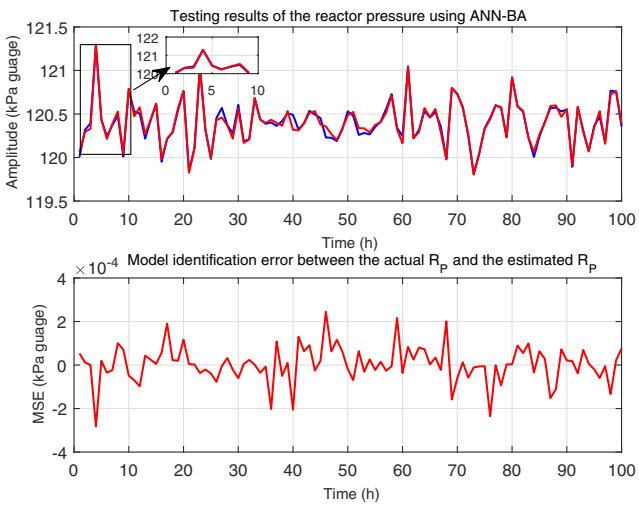

Fig. 22 Testing results of the reactor pressure and the prediction error between the actual R_P and the estimated R_P using ANN-BA

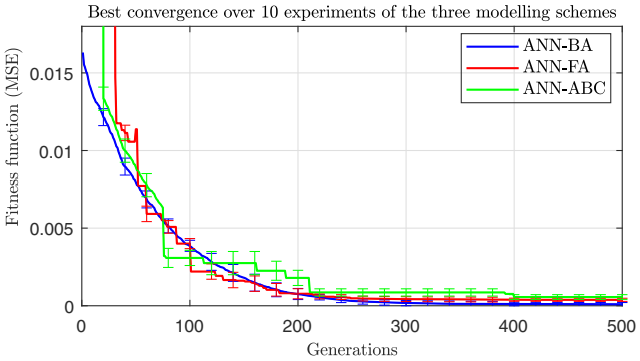

Fig. 23 Convergence curves of ANN-BA, ANN-FA, and ANN-ABC for the reactor temperature subsystem at different MSE values

the reactor temperature subsystem are presented for up to 500 iterations in Fig. 23. The curves are identified by the blue and red color of the ANN-BA, and ANN-FA models, respectively and the ANN-ABC curves are shown by a green line.

The data of the plots in Fig. 23 represents the MSE or the fitness function of each evaluated algorithm as defined by the mean of the sum of the squared errors between the data values created by the models and the equivalent actual data measurements. The error bars represent a unit standard error of the mean of the data presented in the curves.

There is a slight difference between the results of ANN-BA and ANN-FA, which suggests that the significance is lesser than the 63% propositioned by the error bars for each point on the curves. There is a consistent difference between the ANN-FA and ANN-ABC curves at the first 200 iterations which suggests that the significance is greater than the 80% suggested by the error bars for each point on the curve. The error bars of the curves created by ANN-BA and ANN-FA overlap at several points, and the degree of separation between the curves exists at distinct few points along the curves. This implies that there is no considerable deviation occurring in the data patterns for both points of the curves to coincide at that point. However, for a fraction of the curve, this distinction is significant pointing out that the ANN-BA model can achieve better modeling. The error bars for the curves of the models created by ANN-FA and ANN-ABC overlap slightly at the last 150 iterations which means that the difference at these iterations is not significant in a statistical sense. This means that it would not take an extreme deviation that occurs 66% in the data for both points for the curves to coincide at that point. The curve plots of the ANN-BA and ANN-ABC models are clearly separated, and the error bars do not overlap along the curves. Therefore, there is a very high confidence that the difference is remarkable because this degree of separation presents at most points along the curves.

The convergence process of all proposed models resulted in a rapid increase in rapprochement at onset and gradually slowed down to reach an optimal error. Overall, the plot curves in Fig. 23 show that the model structures presented by ANN-BA, ANN-FA, and ANN-ABC have achieved a high degree of accuracy in modeling the reactor temperature sub-process. They each converged to a very small MSE value over the same number of generations. Besides, the convergence plot of ANN-BA reached a small error rate after 300 iterations and improved a little further. It also shows that a very small improvement for ANN-ABC is obtained after 200 iterations (shown in green). There is a further small reduction in this value after 400 iterations.

8.3 Computational results

The trained models captured the main properties of the TE datasets and the variability of the weight parameters so that they were used to constrain the modeling process to identify the unfamiliar output of the evaluated data. Table 5 shows the training and validation MSE values of the developed model schemes.

Table 5 shows that the modeling schemes presented here reported optimal MSE values in the training and identification processes for each model developed. Specifically, it presents at the end of the training an MSE rate of $6.54e-3$ for R_L subsystem using the ANN-BA scheme, compared to a $9.88e-3$ MSE rate for the R_L using the ANN-FA scheme. These values were reported for the same ANN configuration, making a difference in MSE rate of 00734 a small and not significant. Also, there is a small difference, as shown

Table 5 Average mean square training and testing errors arrived at by the proposed approaches for ten experiments for the developed models of the TE industrial process

Method	MSE							
	Training data				Testing data			
	R_L	R_P	R_C	R_T	R_L	R_P	R_C	R_T
ANN-BA	2.54e−3	1.10e−2	2.59e−6	1.10e−6	1.29e−2	9.37e−2	2.67e−5	4.91e−5
ANN-FA	9.88e−3	5.53e−2	8.67e−6	6.86e−6	5.82e−2	1.19e−3	6.96e−5	8.74e−5
ANN-ABC	1.46e−2	7.22e−2	2.37e−5	6.90e−5	9.98e−1	9.52e−1	5.52e−4	9.27e−5

Table 6 Correlation similarity measure between the actual output signal and the corresponding estimated one of the four TE two-phase reactor subsystems using the three proposed modeling schemes

Method	Training correlation (%)				Testing correlation (%)			
	R_L	R_P	R_C	R_T	R_L	R_P	R_C	R_T
ANN-BA	94.167	87.168	99.936	99.998	90.167	84.668	99.236	99.952
ANN-FA	91.777	84.957	99.341	99.941	89.334	82.927	98.907	99.917
ANN-ABC	90.185	82.165	98.404	99.107	86.452	80.365	96.225	98.970

in Table 5, between the testing MSE values of R_C and R_T models reported by ANN-BA, ANN-FA, and ANN-ABC approaches.

To get a deeper look at the obtained results, a statistical analysis was performed by calculating the correlation between the actual output amplitudes and the corresponding estimated one for the three proposed methods: ANN-BA, ANN-FA, and ANN-ABC. The correlation values are shown in Table 6.

Table 6 shows a very large correlation between the predicted and the true R_L, R_P, R_C, and R_T, where the minimum value of 80.365 was reported in the case of R_P during testing the ANN-ABC model. The statistical correlation results are encouraging and impose that the proposed modeling schemes are robust models representation. As can be seen, statistically, the ANN-BA is a little more compact than the ANN-FA, as the correlation measure obtained is, in each case, a slightly upper. As a consequence, ANN-BA is more powerful in all subsystems studied because it produced a high level of accuracy in all TE sub-models.

8.4 A comparison with other reported models in the literature

To obtain a quantitative assessment of the performance of the proposed modeling schemes, we compared our approach with other approaches that modeled the same process. The models were trained on 150 data samples of each variable of the four TE subsystems and tested them on a different set of 150 data samples of each of the four variables of the four corresponding TE subsystems. Table 7 shows training and verification VAF values of the four TE models obtained using the proposed schemes compared to the VAF values of existing TE models for the corresponding TE subsystems. The VAF results, of the existing models, have been reported by the authors.

Table 7 compares the evaluation strategy and the performance accuracy of the models presented in this paper to four competitive modeling methods [2, 3, 11, 28] for the same industrial process. It is clear that a high level of performance is obtained using the proposed models. Moreover, the VAF results of the proposed models are considerably better than the state-of-the-art models such as [3, 28], particularly for R_L and R_P models. This illustrates that the presented models were managed to be a good representation of the TE subsystems. This implies that the preprocessing step of the scaling and filtration processes helped to generate thoroughly stable models towards the end of training and validating phases. The state-of-art models may not be flexible with a variation of the TE subsystems data, whereas it can be inferred from Table 7 that the responses of these models encountered glitches during modeling the reactor level and pressure

Table 7 A comparison in terms of the average VAF results between the proposed modeling approaches and other reported modeling approaches for the TE two-phase reactor for a set of ten experiments

Method	VAF (%)							
	Training data				Testing data			
	R_L	R_P	R_C	R_T	R_L	R_P	R_C	R_T
FL	68.587	46.497	97.705	99.934	–	–	–	–
NNARX	71.978	11.320	99.872	99.817	–	–	–	–
ANFIS	75.174	51.410	99.218	99.939	–	–	–	–
GP	78.107	59.816	95.819	99.873	56.567	28.284	95.761	99.812
ANN-BA	93.366	84.855	99.911	99.989	85.368	79.479	97.427	99.925
ANN-FA	91.656	81.282	98.682	99.911	83.611	78.675	96.181	99.873
ANN-ABC	89.598	79.976	97.548	99.003	81.017	75.682	94.850	98.949

systems. Furthermore, the proposed modeling scheme such as ANN-BA achieved testing VAF values for R_C and R_T models slightly better than the corresponding state-of-the-art models reported by [11]. The difference is respectively about 1.666 and 0.113 which is not that big.

The evaluation results obtained using ANN-BA and ANN-FA are analogous to a relatively small difference compared to that of ANN-ABC. The difference in VAF rate between ANN-BA and ANN-FA schemes is less than 0.9648 on average which is small and insubstantial. These results demonstrate that the training method of ANN using BA or FA is not critical to the performance of the models.

The results in Tables 5, 6, and 7 reveal that BA, FA, and ABC algorithm is efficacious in adjusting the weights of ANNs, learned to successfully navigate the course on the majority of test runs and often reaching optimal MSE and VAF values. All in all, the overall results prove that the proposed modeling techniques have produced valuable and adaptable models in representing the behavior of the TE two-phase chemical reactor and further are capable of obtaining good mathematical models for other complex industrial and control problems.

9 Conclusions and future work

This research deals with a common problem that arises during the design of artificial neural networks (ANNs): accuracy of the ANN. In this sense, the fitness functions take into account the mean square error (MSE) and the classification error. The fitness function is used to evaluate each solution and find the best design. The evolutionary computation algorithms are used to get the set of weights of the ANN with an aim to resolve a prediction problem of the Tennessee Eastman (TE) reactor process. Furthermore, performance comparison between the proposed ANN-BA, ANN-FA, and ANN-ABC is done versus several classical learning methods, which are designed manually using the well-known Back-Propagation and Levenberg-Marquardt learning algorithms along with the fuzzy logic and neuro-fuzzy approaches. Also, the accuracy is assessed using the MSE and variance account for (VAF) criteria. A high level of predictability is shown, where the reported MSE and VAF rates of all the developed models of the TE reactor subsystems demonstrated a high degree of validity and accuracy. The results give a positive indication of the potentiality offered by the evolutionary algorithms in training ANNs to present a reliable and effective approach for modeling the dynamic aspect of any nonlinear industrial process. They further increased the level of adaptability of ANNs to produce an efficient modeling approach.

The use of a new simulation model based on ANN with evolutionary optimization algorithms in the development of an identification system for the TE reactor subsystems is of high value for future research. Further work is needed to assess the appropriateness of the proposed approach to other publicly available real industrial benchmark control datasets. Further study is needed to evaluate the proposed approach with other evaluation objective functions. Expansion of the proposed algorithm to other harmonic optimization algorithms would be valuable.

Compliance with ethical standards

Conflict of interest The authors declare they have no conflict of interest.

References

1. Tennessee eastman challenge archive. depts.washington.edu/control/LARRY/TE/download.html. Accessed 17 Jan 2018
2. Al-Hiary H, Braik M, Sheta A, Ayesh A (2008) Identification of a chemical process reactor using soft computing techniques. In: IEEE International conference on fuzzy systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE, pp 845–853
3. Al-Hiary H, Sheta A (2006) Identification and model predictive controller design of the Tennessee Eastman chemical process using ANN. In: Proceeding of 4th international multi-conference on computer science and information technology (CSIT 2006), Amman, Jordan, April 5-7, vol 2, pp 189–198
4. Bathelt A, Ricker N, Jelali M (2015) Revision of the Tennessee Eastman process model. IFAC-PapersOnLine 48(8):309–314
5. Braik M, Sheta A, Arieqat A (2008) A comparison between GAs and PSO in training ANN to model the TE chemical process reactor. In: Proceedings of the AISB 2008 convention in communication, interaction and social intelligence, vol 1, p 24
6. Chen H, Tiňo P, Yao X (2014) Cognitive fault diagnosis in Tennessee Eastman process using learning in the model space. Comput Chem Eng 67:33–42
7. Chen TB, Dong YL, Jiao YC, Zhang FS (2006) Synthesis of circular antenna array using crossed particle swarm optimization algorithm. J Electromagn Waves Appl 20(13):1785–1795
8. Dong J, Zhang K, Huang Y, Li G, Peng K (2015) Adaptive total pls based quality-relevant process monitoring with application to the Tennessee Eastman process. Neurocomputing 154:77–85
9. Downs JJ, Vogel EF (1993) A plant-wide industrial process control problem. Comput Chem Eng 17(3):245–255
10. Eldem V, Yildizbayrak N (1988) Parameter and structure identification of linear multivariable systems. Automatica 24(3):365–373
11. Faris H, Sheta A (2013) Identification of the Tennessee Eastman chemical process reactor using genetic programming. Int J Adv Sci Technol 50:121–140
12. Ge Z (2017) Review on data-driven modeling and monitoring for plant-wide industrial processes. Chemometrics and Intelligent Laboratory Systems
13. Haykin S (1994) Neural networks: a comprehensive foundation. Prentice Hall PTR
14. Jia Q, Zhang Y (2016) Quality-related fault detection approach based on dynamic kernel partial least squares. Chem Eng Res Des 106:242–252
15. Jockenhövel T, Biegler LT, Wächter A (2003) Dynamic optimization of the Tennessee Eastman process using the OptControlCentre. Comput Chem Eng 27(11):1513–1531

16. Karaboga D (2010) Artificial bee colony algorithm. Scholarpedia 5(3):6915
17. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. J Global Optim 39(3):459–471
18. Li W, Hongbo S (2014) Improved kernel pls-based fault detection approach for nonlinear chemical processes. Chin J Chem Eng 22(6):657–663
19. Liu D, Michalski KA (2016) Comparative study of bio-inspired optimization algorithms and their application to dielectric function fitting. J Electromagn Waves Appl 30(14):1885–1894
20. Ljung L (1987) Theory for the user. Prentice Hall
21. Mish MJR (2009) Swarm intelligence techniques and its applications in water resources management ISH. J Hydraul Eng 15(1):151–169
22. Reddy MJ, Kumar DN (2007) An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design. Eng Optim 39(1):49–68
23. Ricker N (1995) Optimal steady-state operation of the Tennessee Eastman challenge process. Comput Chem Eng 19(9):949–959
24. Ricker N, Lee J (1995) Nonlinear modeling and state estimation for the Tennessee Eastman challenge process. Comput Chem Eng 19(9):983–1005
25. Ricker N, Lee J (1995) Nonlinear model predictive control of the Tennessee Eastman challenge process. Comput Chem Eng 19(9):961–981
26. Rumelhart DE, Hinton GE, Williams RJ (1985) Learning internal representations by error propagation. Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science
27. Schalkoff RJ (1997) Artificial neural networks, vol 1. McGraw-Hill, New York
28. Sheta A (2005) Modeling the Tennessee Eastman chemical reactor using fuzzy logic. ISE Book Series on Fuzzy System Engineering-Theory and Practice published by Nova Science
29. Sheta AF, Braik M, Al-Hiary H (2009) Identification and model predictive controller design of the Tennessee Eastman Chemical Process using ANN. In: Proceedings of the international conference on artificial intelligence (ICAI'09), July 13-16, USA, vol 1, pp 25–31
30. Sheta AF, Braik M, Öznergiz E, Ayesh A, Masud M (2013) Design and automation for manufacturing processes: an intelligent business modeling using adaptive neuro-fuzzy inference systems. In: Business intelligence and performance management. Springer, pp 191–208
31. Wang G, Tang W, Xia J, Chu J, Noorman H, Gulik WM (2015) Integration of microbial kinetics and fluid dynamics toward model-driven scale-up of industrial bioprocesses. Eng Life Sci 15(1):20–29
32. Webb B (2002) Swarm intelligence: from natural to artificial systems. Connect Sci 14(2):163–164
33. Wei Y, Qiu J, Lam HK, Wu L (2017) Approaches to t–s fuzzy-affine-model-based reliable output feedback control for nonlinear ito stochastic systems. IEEE Trans Fuzzy Syst 25(3):569–583
34. Yang XS (2009) Firefly algorithms for multimodal optimization. In: International symposium on stochastic algorithms. Springer, pp 169–178
35. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. Int J Bio-Inspired Comput 2(2):78–84
36. Yang XS (2010) A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization (NICSO 2010), pp 65–74
37. Yin S, Wang G, Gao H (2016) Data-driven process monitoring based on modified orthogonal projections to latent structures. IEEE Trans Control Syst Technol 24(4):1480–1487
38. Zweiri YH, Seneviratne LD, Althoefer K (2005) Stability analysis of a three-term backpropagation algorithm. Neural Netw 18(10):1341–1347